SimPowerSystems

For Use with Simulink®

Hydro-Québec

TransÉnergie Technologies

- Modeling
- Simulation
- Implementation

Conversion Guide

Version 3

The MathWorks

How to Contact The MathWorks:

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup

`support@mathworks.com`	Technical support
`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information

508-647-7000 Phone

508-647-7001 Fax

The MathWorks, Inc. Mail
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

SimPowerSystems Conversion Guide

Printing History: September 2003 Online only Special for Version 3 (Release 13SP1)

Contents

Converting Version 2 Models

Converting Version 2 Models

You can convert power system models created with previous versions of the block library, from SimPowerSystems 2.3 or Power System Blockset 2, to the new block library and Physical Modeling interface of SimPowerSystems 3.0. This chapter explains how to update your models from Version 2 to Version 3 and the problems you might encounter.

This chapter includes online links to the old SimPowerSystems 2.3 documentation.

Converting Your Old Models to Version 3

When you open an old model in SimPowerSystems 3.0 for the first time, its blocks are linked to the block library **powerlib2** of SimPowerSystems 2.3 or Power System Blockset 2. Consequently your models are not automatically updated with the new block library **powerlib** of Version 3.0.

To take full advantage of the new SimPowerSystems, it is strongly recommended that you convert your models with the `psbupdate` command to the new interface and block library. This automatic conversion accomplishes most of what you need to run your models with **powerlib**. This section explains the model conversion steps.

Running Version 2 Models

For your convenience, you can still run your old models as they are and obtain the same simulation results you would get with previous releases.

SimPowerSystems 3.0 includes the old **powerlib2** block library. You can open this library by entering `powerlib2` at the command line.

Caution You cannot put Version 2 and Version 3 blocks in the same model file.

Accessing the SimPowerSystems Version 2 Documentation

As you work with models built with blocks from the **powerlib2** library in SimPowerSystems Version 2.3 or Power System Blockset Version 2, you might want to consult the Version 2 documentation, either the online form or the printed *Power System Blockset User's Guide,* Version 2.

Online users can access the HTML or PDF version of the SimPowerSystems 2.3 documentation by clicking the links below. The old documentation does not include the old Release Notes or the old product page (Web) link. Consult the Version 3.0 documentation for updated release notes and product page.

- Online HTML version of the SimPowerSystems 2.3 User's Guide
- Online PDF version of the SimPowerSystems 2.3 User's Guide

Using the psbupdate Command

The `psbupdate` command converts models built with Version 2 blocks into the new Physical Modeling block diagram interface and library of SimPowerSystems 3.0. The conversion happens in three steps:

1. New **powerlib** blocks replace old **powerlib2** blocks. The block ports are sometimes reordered.

2. Physical Modeling connection lines replace Simulink signal lines, if those lines connect the new Physical Modeling electrical terminal ports on the converted blocks.

   Special JunctionPoint blocks are inserted at line branch points. Subsystem ports are converted, if necessary, to the new Connection Port blocks. See "Terminal Ports for the Electrical Domain of Physical Modeling" on page 5 and "Special Conversion Blocks and Retired Version 2 Blocks" on page 6.

   The conversion can fail to convert and reconnect some connection lines. See "Repairing Incomplete Connection Lines" on page 7.

3. The `psbupdate` function changes the name of your model to `<oldmodelname>_updated` and saves it under this name so you can easily refer to the original model if you need to reconnect broken lines.

Starting the Conversion

The `psbupdate` command is easy to use. Open the model you want to convert. Then invoke the command by entering

```
psbupdate('Name_Of_The_Model_To_Convert')
```

at the command line.

`psbupdate` converts every Version 2 block of your model to SimPowerSystems 3.0 blocks. It also converts any subsystem that uses Version 2 blocks. The old blocks need to be linked to **powerlib2** prior to the conversion. Otherwise they are not converted. If your model contains blocks or subsystems linked to a user-defined library, you must update this library first, then update your model. Alternatively, you can break your model's link to the user-defined library, update your model, and then propagate the changes for these linked blocks to the user-defined library.

The `psbupdate` command ignores regular Simulink blocks or Simulink subsystems while it updates your model. Consult the Simulink documentation on working with block libraries if you need to update those portions of your model.

The new Version 3.0 blocks use the new Physical Modeling terminal ports. `psbupdate` deletes the Simulink signal lines connecting the old blocks and replaces them with Physical Modeling electrical connection lines, if they are needed to interconnect the new terminal ports. The replacement of Simulink lines is the most critical task of the conversion process, and the conversion sometimes fails to complete every line.

The Generated Conversion Report

The `psbupdate` command displays a report in the command window each time you convert a model. It reports

- The blocks that `psbupdate` failed to convert because of broken links to the block library **powerlib2**
- The Simulink signal lines that are too short to be replaced
- The electrical connection lines that are not properly reconnected

In addition to the report, `psbupdate` identifies the two kinds of connection line problems in the model:

- For each line that is too short to be replaced, a special problem-identifier block named `Problem #1`, `Problem #2`, etc., is added to the model at the location where the line should go.
- Each of the lines that `psbupdate` failed to reconnect is identified by a tag: `unconnected line #1`, `unconnected line #2`, etc.

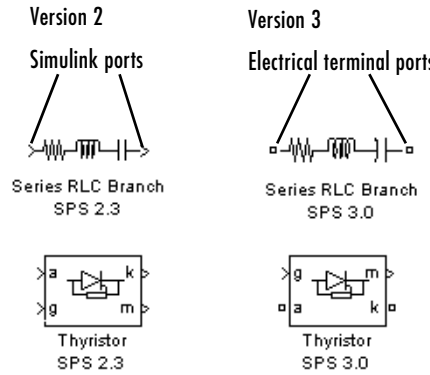
Block Changes Since Version 2.3

There are no differences in block parameters between a **powerlib2** block and its **powerlib** replacement. When you update your model, the parameters you set or entered in the blocks of your old model are automatically transferred into the updated model. This section explains the differences between **powerlib2** and **powerlib** blocks.

Terminal Ports for the Electrical Domain of Physical Modeling

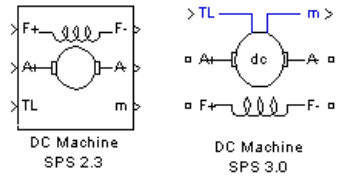
The main difference between the blocks of the old **powerlib2** library and the new **powerlib** library lies in their electrical ports. SimPowerSystems 3.0 introduces a new type of block port, the terminal port, specific to the electrical domain of Physical Modeling and different from the Simulink input and output ports. It is represented by a small white square □ .

In the previous versions of SimPowerSystems and Power System Blockset, the electrical block ports were modeled by Simulink input and output ports. The distinction between Simulink ports and electrical terminal ports is now clearly defined. For example, the old and new versions of the Series RLC Branch and Thyristor blocks show the difference in ports:



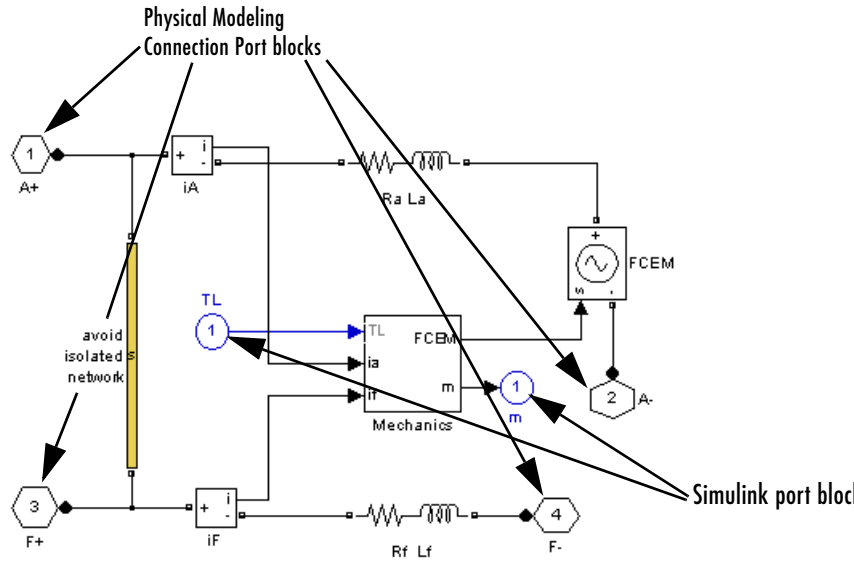
Notice the new ordering of the Simulink and the electrical ports. The Simulink input and output ports are always in first position, and the electrical ports follow them. In the Thyristor block, the Simulink ports named a and k are converted into electrical ports (small white squares) and are now the electrical input and output ports of the block.

This change applies to every Version 2 block that has both Simulink and electrical terminal ports. The reordering of the ports also applies to any subsystem that uses Simulink and electrical ports. This is the case, for example, in the DC Machine block of the Machines library:



Special Conversion Blocks and Retired Version 2 Blocks

The new Connection Port block, placed in a subsystem, models an electrical terminal port on the subsystem's outside. This new block looks different from Simulink subsystem port blocks. Look under the mask of the DC Machine block. The Inport block implements the TL input of the DC Machine, and the Outport block models the output port named m:



Four Connection Ports model the four electrical terminal ports of the DC Machine block. Notice also that the two types of blocks have separate port numberings for the ordering of their respective ports.

Previous versions of the SimPowerSystems and Power System Blockset libraries have the Bus Bar block and the two special T and L Connector blocks to interconnect other blocks. In the Version 3.0 library, these blocks no longer exist. However, the `psbupdate` command replaces them with a similar new block, JunctionPoint. `psbupdate` uses this block only to preserve the appearance of your old models. You can remove the JunctionPoint blocks after conversion and replace them with branched connections lines.

For building entirely new Version 3 models, you have no need for the Connection Port and JunctionPoint blocks. But you can access them in the hidden **power_utility** block library by entering `power_utility` at the command line.

Repairing Incomplete Connection Lines

In some cases, `psbupdate` only converts certain old Simulink lines and fails to convert other lines that should have been changed. The conversion might also fail to reorder the block ports on all converted blocks. When this problem arises, some lines or blocks are left unconnected in the converted model.

This connection line conversion problem occurs when two or more blocks are very close to one another. In such cases, some Simulink lines are very short, and `psbupdate` misses them, failing to replace the Simulink signal lines with electrical connection lines.

Inspecting Converted Models for Broken Connection Lines

This potential conversion failure means that you need to visually inspect every automatically converted model for completeness and manually reconnect any broken lines. To assist you in manually completing the conversion, `psbupdate` reports each conversion failure in the automatic conversion report and locates the problem in the model. See “The Generated Conversion Report” on page 4.

Inspecting Converted Subsystems for Broken Connection Lines

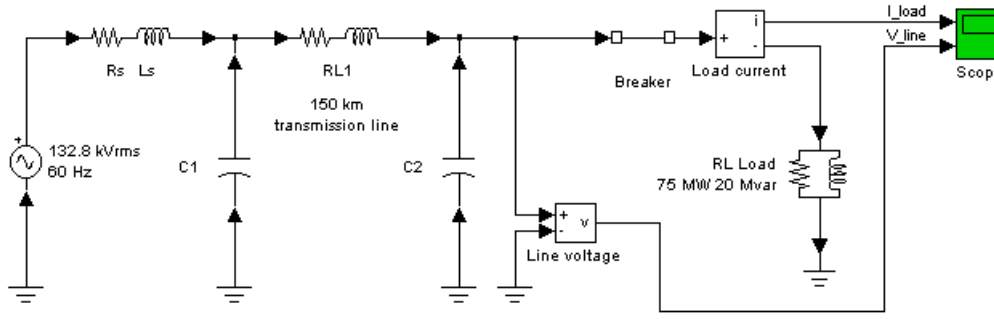
For models with interconnected subsystems, `psbupdate` can also fail to convert some or all of the Simulink lines between subsystems. If such a failure occurs with your model, you must manually delete these lines (converted to dotted red lines) and reconnect the subsystems with electrical connection lines. An

example of such a problem is given in the following section, “Example of a Complex Model Conversion” on page 18.

Note When manually completing a converted model with nested subsystems, it is essential that you fix the innermost subsystems first. Then work up the subsystem hierarchy and be sure to identify improperly ordered and disconnected subsystem ports.

Example of a Simple Model Conversion

Try this conversion of a simple Version 2 circuit containing two T Connector blocks. Open the `psbtransient` demo of SimPowerSystems 2.3. The blocks in the model are linked to the **powerlib2** library.



Now update the model to the Version 3.0 by entering the `psbupdate` command:

```
psbupdate('psbtransient')
```

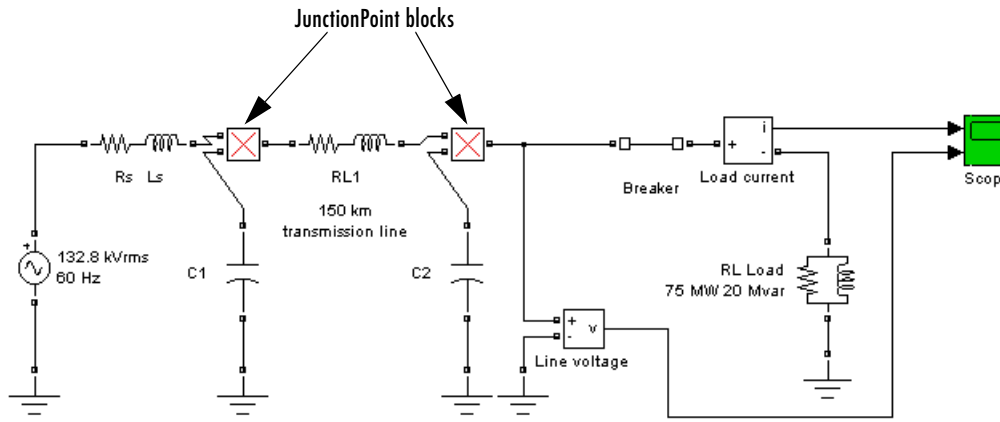
Conversion Results

The automatic report appears in the command window:

```
------------------------------------------------------------------
Conversion of "psbtransient" into SimPowerSystems 3.0

Conversion Summary

Converted model name: psbtransient_updated
Number of unconverted blocks: 0
Number of lines too short to be replaced: 0
Number of lines not reconnected: 0
Check the resulting new circuit to ensure that all the blocks and
lines are properly connected.
------------------------------------------------------------------
```

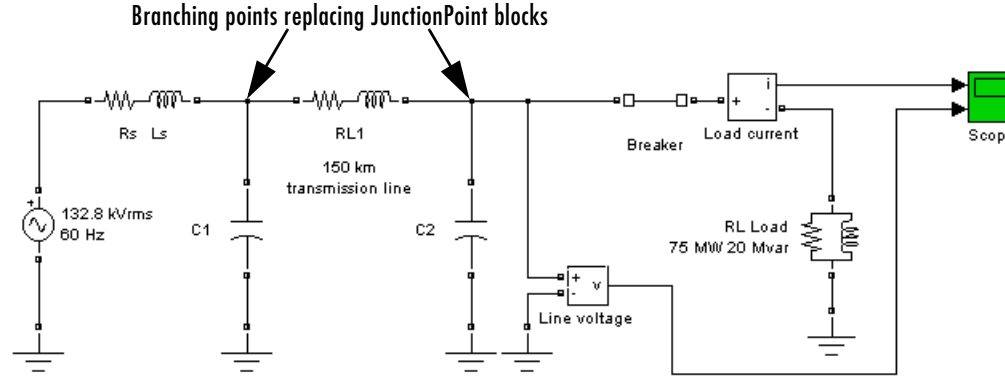
The converted model is shown in the following figure. Note that the two T Connector blocks are now replaced by two JunctionPoint blocks. The

JunctionPoint blocks are colored red in the model (the blocks with a red **X** displayed as the block icon):



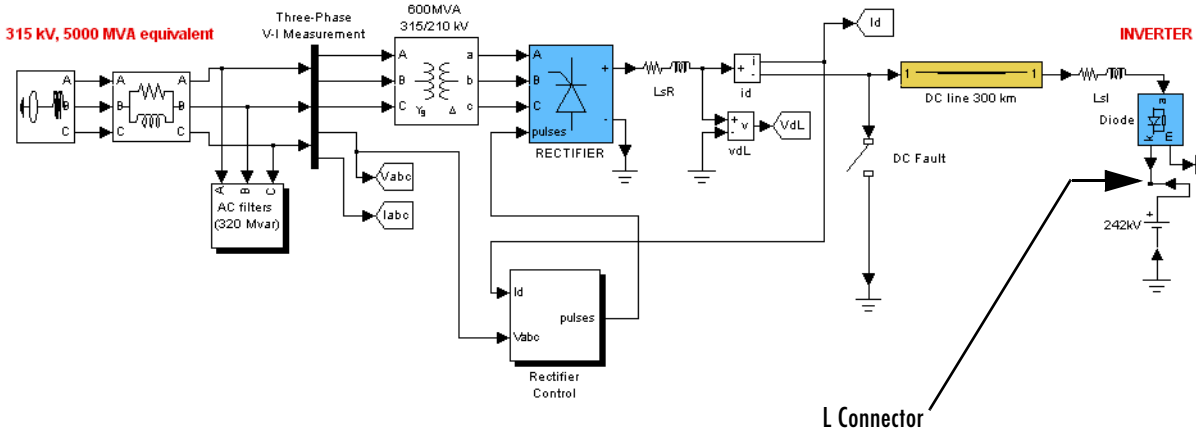
Optional Manual Editing

You can run the converted circuit as is and should give the same simulation results as the original model. But you can manually edit the model by deleting the two JunctionPoint blocks and reconnecting the appropriate lines by branching. The conversion of the original model is then complete.



Example with Broken Connection Lines

This example illustrates the conversion failure that results when you have two or more blocks very close to one another. The signal lines are very short, and `psbupdate` cannot replace these lines correctly. A modified version of the `psbhvdc` demo of Version 2 illustrates this problem. Here is the original model:



Note the connection point between the Voltage Measurement block named vdL, the Series RLC Branch block named LsR, and the Current Measurement block named id: the branched connection line is very close to the output port of the LsR block. Note also the close connection of the L Connector block and theDC Voltage Source block 242kV.

Update the model by entering the `psbupdate` command:

```
psbupdate('psbhvdc')
```

Conversion Results

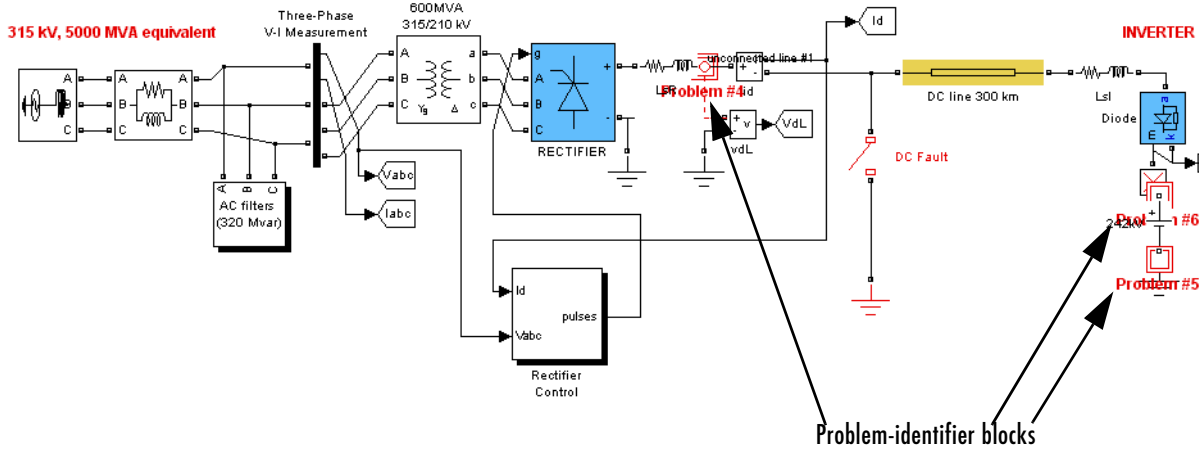
The following report appears in the command window:

```
------------------------------------------------------------------
Conversion of "psbhvdc" into SimPowerSystems 3.0

Block problem #1 in psbhvdc_updated/Subsystem1: The line is too
short to be replaced.
```

```
Block problem #2 in psbhvdc_updated/Subsystem1: The line is too
short to be replaced.
Block problem #3 in psbhvdc_updated/Subsystem1: The line is too
short to be replaced.
Block problem #4 in psbhvdc_updated: The line is too short to be
replaced.
Block problem #5 in psbhvdc_updated: The line is too short to be
replaced.
Block problem #6 in psbhvdc_updated: The line is too short to be
replaced.
Unconnected line #1 in psbhvdc_updated: PSBUPDATE was not able to
reconnect the line.

Conversion Summary

Converted model name: psbhvdc_updated
Number of unconverted blocks: 0
Number of lines too short to be replaced: 6
Number of lines not reconnected: 1
Check the resulting new circuit to ensure that all the blocks and
lines are properly connected.
------------------------------------------------------------------
```

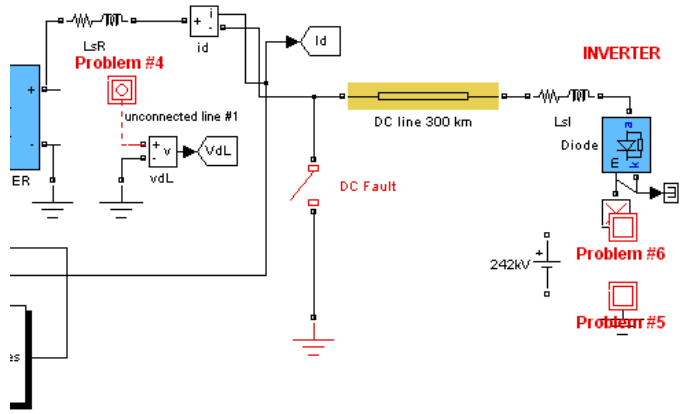
The report identifies six lines too short to be replaced and left unconverted. Consequently some blocks remain unconnected. From the top-level window of the model, we can see three of them. They are identified by three problem-identifier blocks (red squares).

The converted model looks like the following figure. The `psbupdate` command cannot reconnect the LsR block to the measurement blocks and the DC voltage source to the JunctionPoint and the Ground blocks.



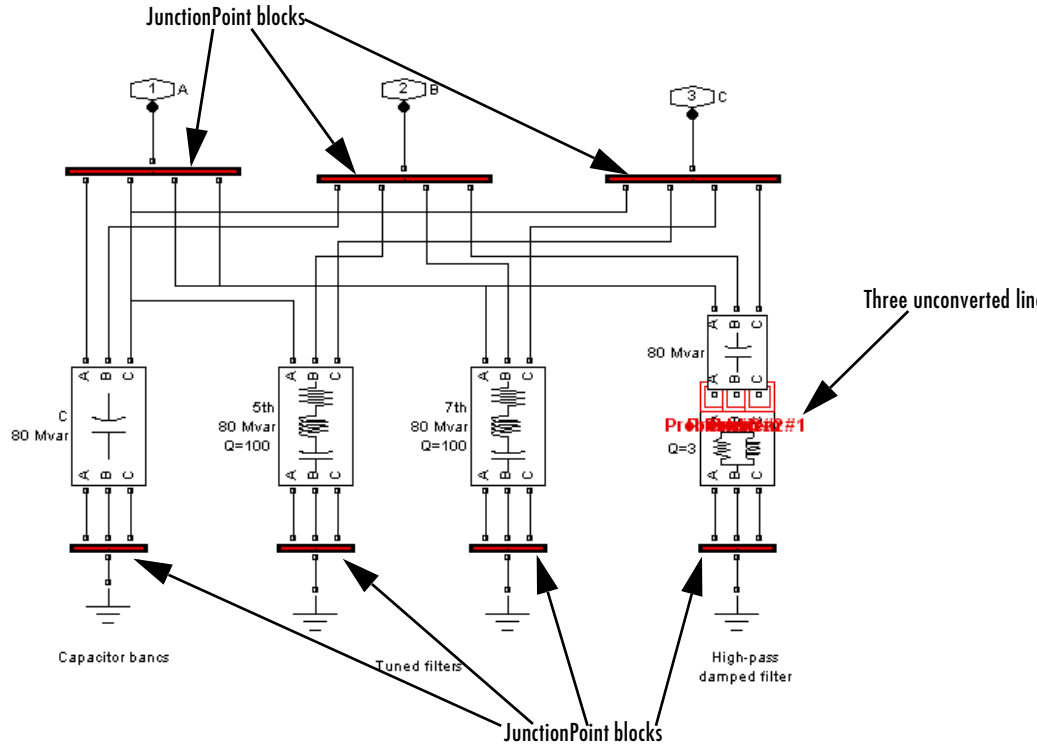
Reconnecting Disconnected Blocks

The report also indicates that one line is converted but not properly reconnected. This line is named `unconnected line #1` and is recognizable by the empty red circle at one endpoint. You can verify that the connections are not complete by moving the LsR and id blocks and the 242kV block from their original positions. You must manually reconnect the unconnected line and blocks and delete the problem-identifier blocks. You can also move the blocks back to their original locations. Compare the following manually edited diagram to the automatically converted model diagram.

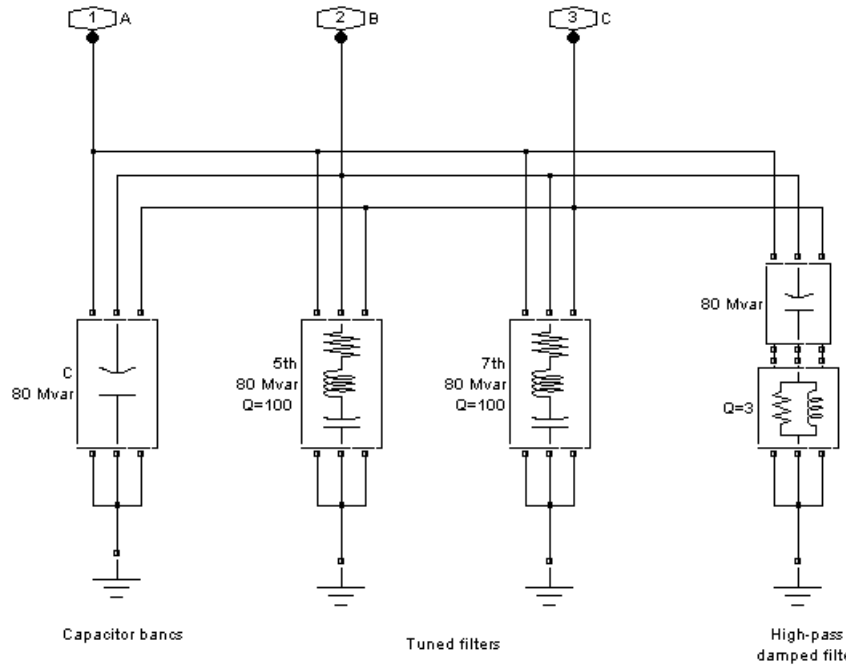


Reconnecting Broken Lines in Subsystems

The rectifier control subsystem is a block that contains no SimPowerSystems blocks. Consequently `psbupdate` ignores it, and the block remains unchanged. Now look inside the AC filters (320 Mvar) subsystem to see if its conversion succeeded. This subsystem contains many Bus Bar blocks to connect the other blocks. The conversion function replaces all the Bus Bar blocks with JunctionPoint blocks. This subsystem contains the three other lines that `psbupdate` failed to replace.



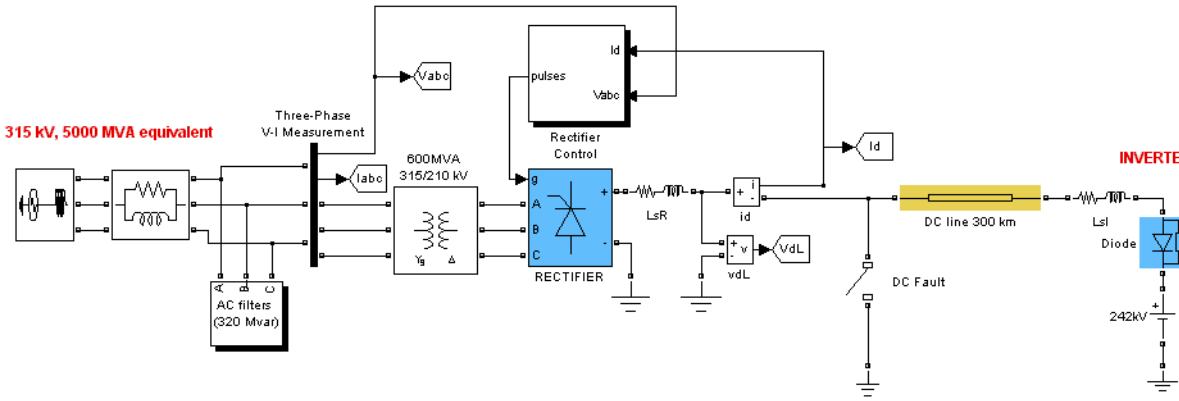
You can reconnect the two blocks and run the simulation. But the diagram would be easier to understand without the JunctionPoint blocks, so you should delete them and reconnect the remaining blocks with branched lines while respecting their phases as you reconnect. The new diagram for this subsystem should look something like the following figure.



Automatic Block Port Reordering

The model conversion also illustrates how a block's ports are reordered. In the original model, at the top level, the three electrical terminal outputs (Phase A, Phase B, and Phase C) are outputs 1, 2, and 3, respectively, and the two measurement outputs of the Three-Phase V-I Measurement block are outputs 4 and 5, respectively. The new version of the block has the measurement outputs as output 1 and output 2, while the three electrical terminal ports are now outputs 3, 4, and 5.

The `psbupdate` command reconnects the blocks. But notice that, because of the reordering of the block ports, the converted diagram looks somewhat disorganized. You can leave the diagram as it is, or you might want to rearrange the blocks that seem out of place so the diagram resembles the following figure. After the conversion and manual editing, the updating of the original model is complete.

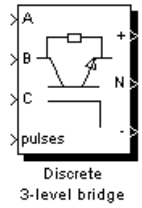

315 kV, 5000 MVA equivalent
Three-Phase
V-I Measurement
Vabc
Iabc
600MVA
315/210 kV
Id
pulses
Vabc
Rectifier
Control
g
A
B
C
RECTIFIER
LsR
id
vdL
VdL
Id
DC line 300 km
INVERTER
Lsl
Diode
242kV
DC Fault
A
B
C
AC filters
(320 Mvar)

Example of a Complex Model Conversion

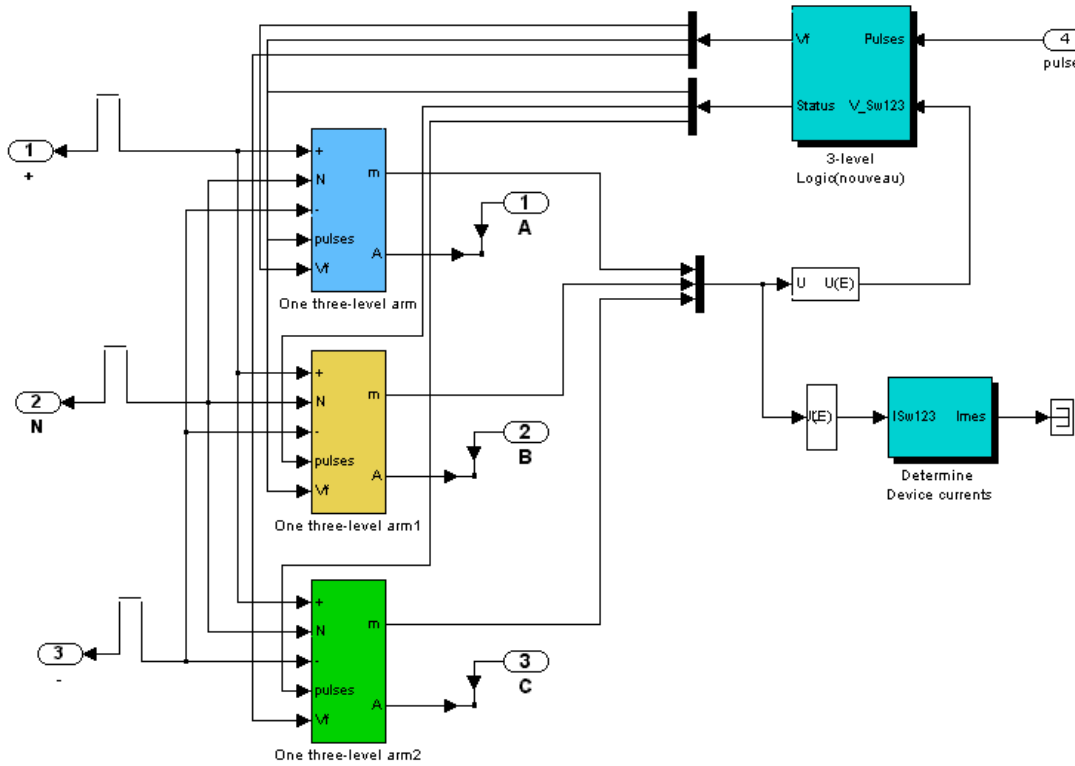
This last example, converting a hypothetical block that models a three-level bridge, illustrates the conversion of a complex model containing nested subsystems and T and L Connector blocks. The `psbupdate` command encounters difficulties with reconnecting the Version 3 subsystems in this model.

Examining the Version 2 Model

Suppose that you created a Version 2 block that models a three-level bridge by using the Ideal Switch blocks as the power electronic device. For each arm of the bridge, you could have created a subsystem containing the ideal switches, the controlled voltage sources that model the Vf voltage characteristic, and the device snubbers represented by Series RLC Branch blocks. Your block might also have included the commutation logic that controls the Ideal Switch block commutations. You might have created a mask and an icon for this block, for example, like this:

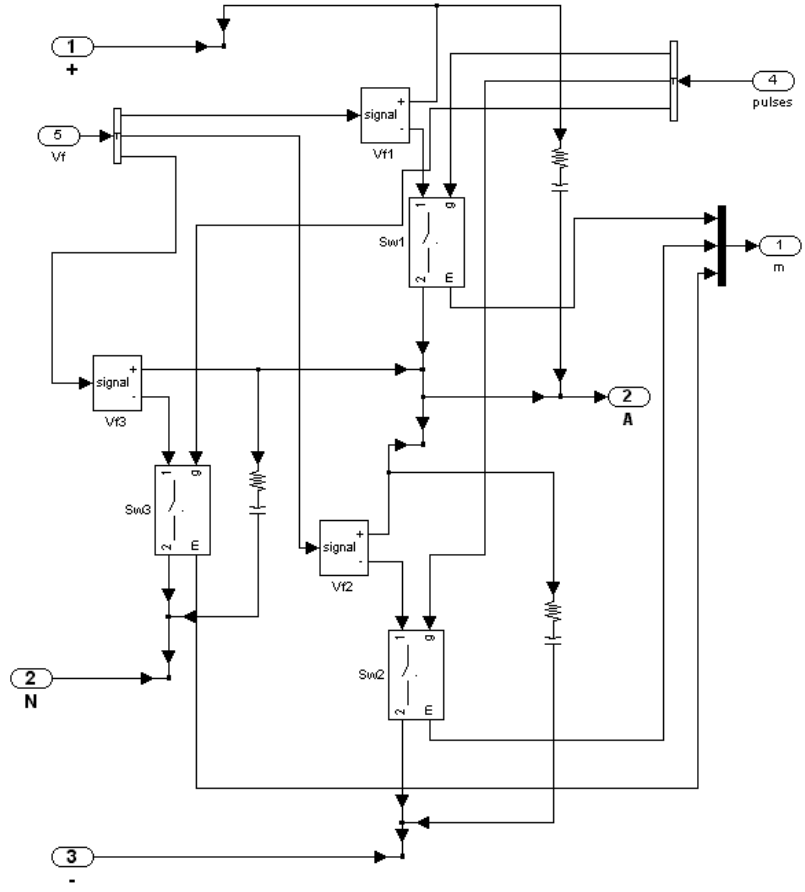


Here is what the Version 2 model's top subsystem might look like under this block's mask:



The 3-Level Logic and Determine Device Currents subsystems in this example consist of Simulink blocks only. `psbupdate` would ignore them during model conversion. The subsystem that models one arm of the bridge might look like

the following figure. Note the presence of many T and L Connectors interconnecting the Version 2 blocks.



Converting the Block to Version 3

If your block were stored in a model named `my3levelbridge`, you could use the `psbupdate` command to update your block by entering

```
psbupdate('my3levelbridge')
```

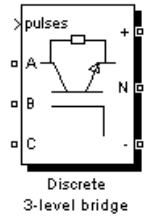
at the command line.

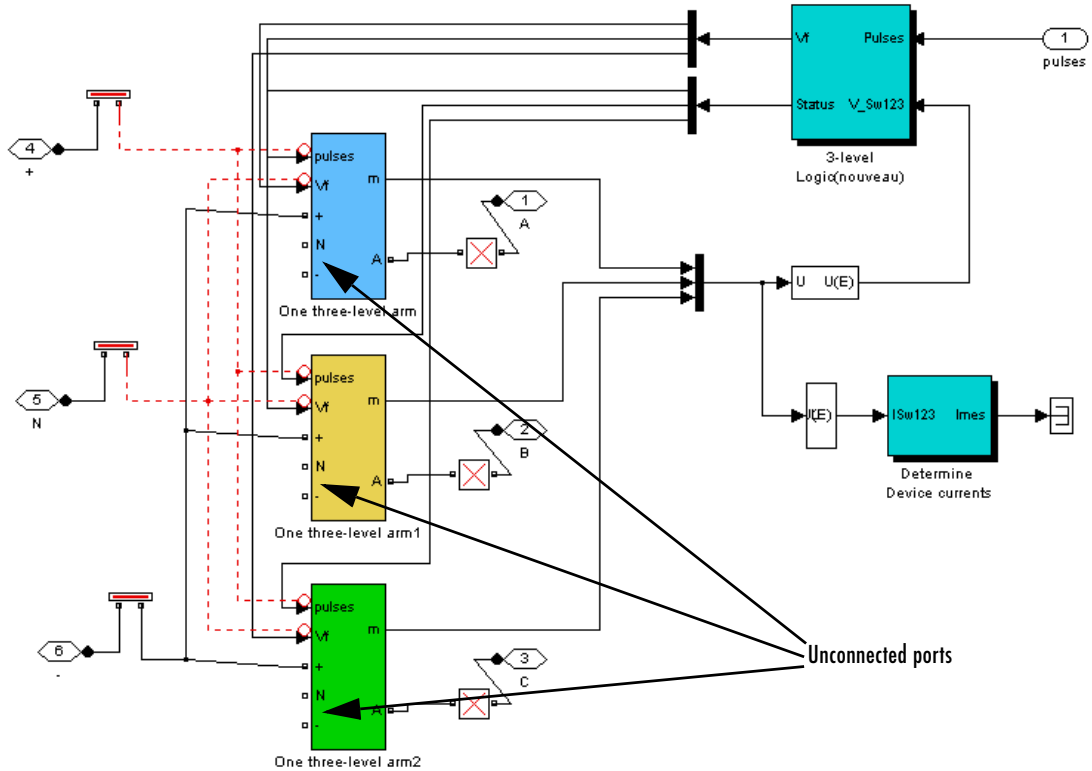
Conversion Results

The automatic report provides the following information:

```
------------------------------------------------------------------
Conversion of "my3levelbridge" into SimPowerSystems 3.0

Block problem #1 in my3levelbridge_updated/Discrete 3-level
bridge/One three-level arm: The line is too short to be replaced.
Block problem #2 in my3levelbridge_updated/Discrete 3-level
bridge/One three-level arm1: The line is too short to be replaced.
Block problem #3 in my3levelbridge_updated/Discrete 3-level
bridge/One three-level arm2: The line is too short to be replaced.
Unconnected line #1 in my3levelbridge_updated/Discrete 3-level
bridge: PSBUPDATE was not able to reconnect the line.
Unconnected line #2 in my3levelbridge_updated/Discrete 3-level
bridge: PSBUPDATE was not able to reconnect the line.
Unconnected line #3 in my3levelbridge_updated/Discrete 3-level
bridge: PSBUPDATE was not able to reconnect the line.
Unconnected line #4 in my3levelbridge_updated/Discrete 3-level
bridge: PSBUPDATE was not able to reconnect the line.
Unconnected line #5 in my3levelbridge_updated/Discrete 3-level
bridge: PSBUPDATE was not able to reconnect the line.
Unconnected line #6 in my3levelbridge_updated/Discrete 3-level
bridge: PSBUPDATE was not able to reconnect the line.
Unconnected line #7 in my3levelbridge_updated/Discrete 3-level
bridge/One three-level arm: PSBUPDATE was not able to reconnect
the line.
Unconnected line #8 in my3levelbridge_updated/Discrete 3-level
bridge/One three-level arm2: PSBUPDATE was not able to reconnect
the line.

Conversion Summary

Converted model name: my3levelbridge_updated
Number of unconverted blocks: 0
Number of lines too short to be replaced: 3
Number of lines not reconnected: 8
Check the resulting new circuit to ensure that all the blocks and
lines are properly connected.
------------------------------------------------------------------
```

The converted block and the top-level diagram are shown in the following figure. The A, B, C, +, N, and – input and output ports are converted into electrical terminal ports, and the Simulink input port named pulses is now in first position on the left side of the block:

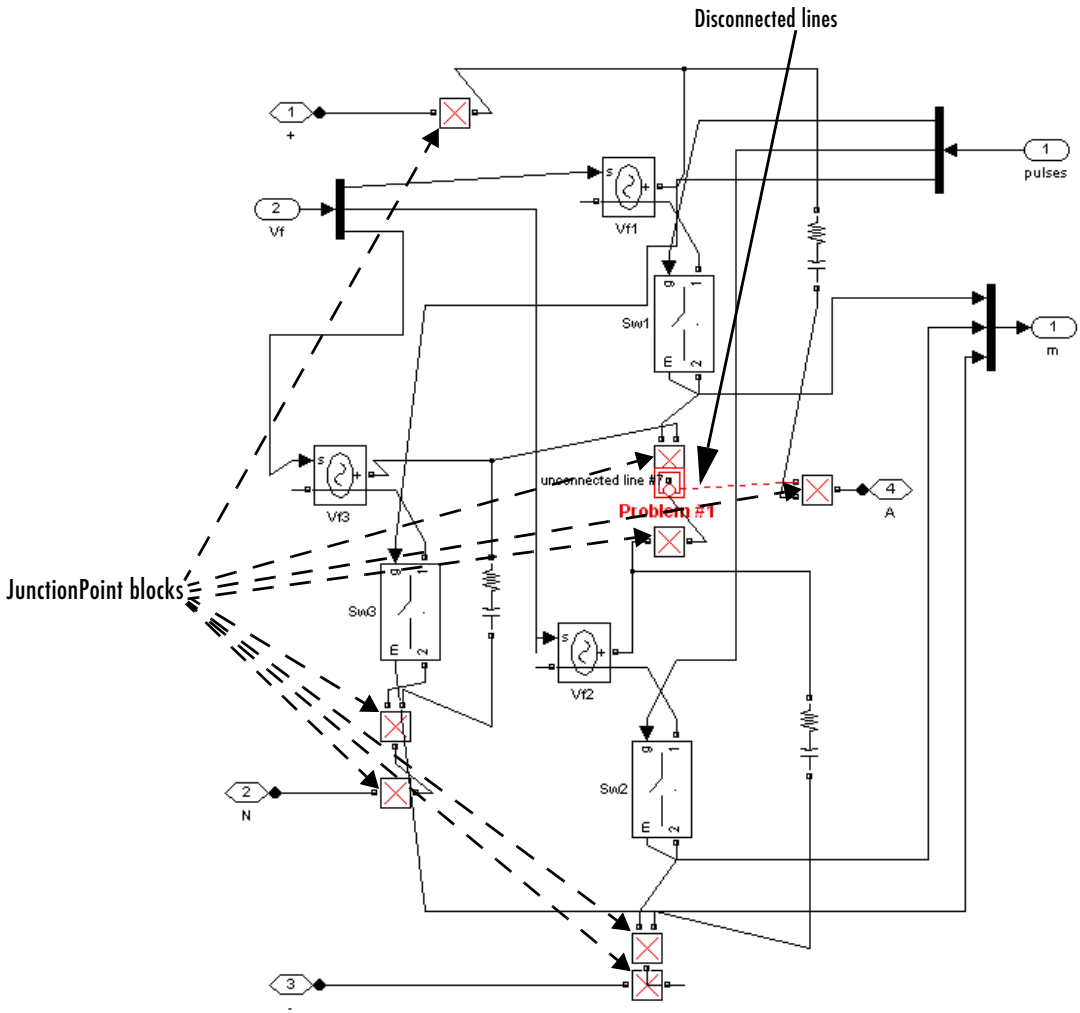




Manually Completing the Subsystem Reconnections

Before you start to fix this diagram, you must fix the innermost subsystems first because the `psbupdate` command might have failed to correctly reorder their electrical ports. (Note the unconnected ports in the diagram above.)

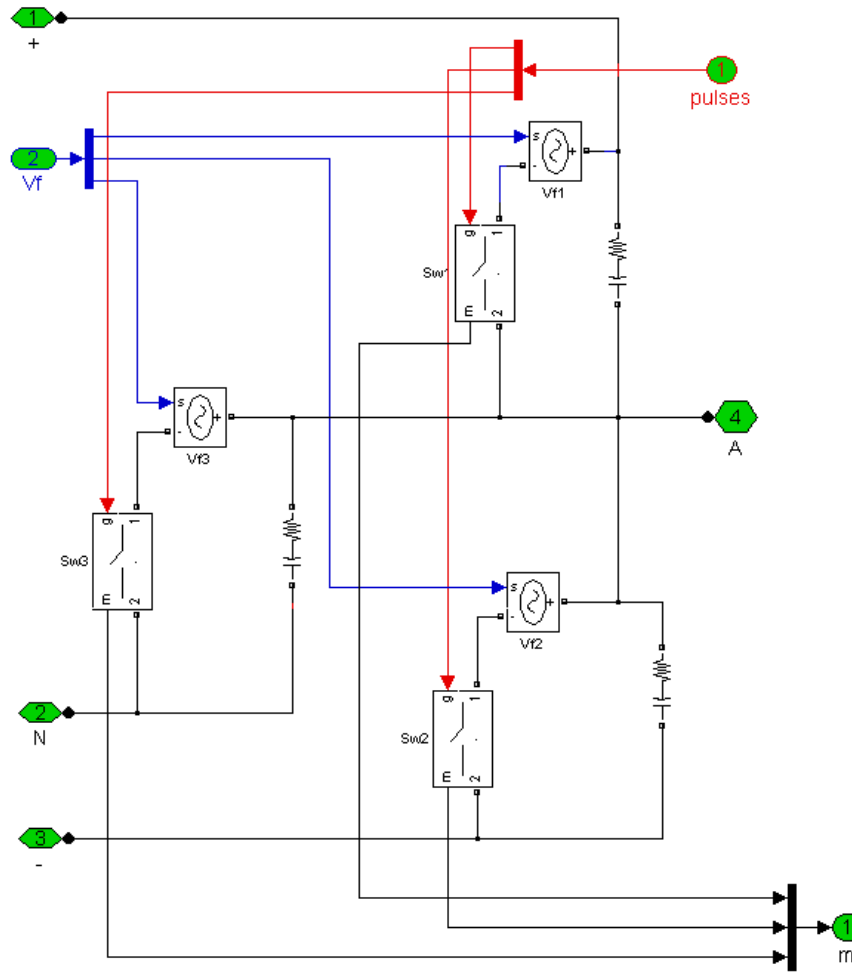
Look under the One three-level arm mask and locate the following diagram:



The updated subsystem appears very disorganized because JunctionPoint blocks have replaced the T and L Connector blocks and the ports of the Ideal Switch blocks have been reordered. But the conversion is functionally perfect except for one line that is not correctly reconnected.

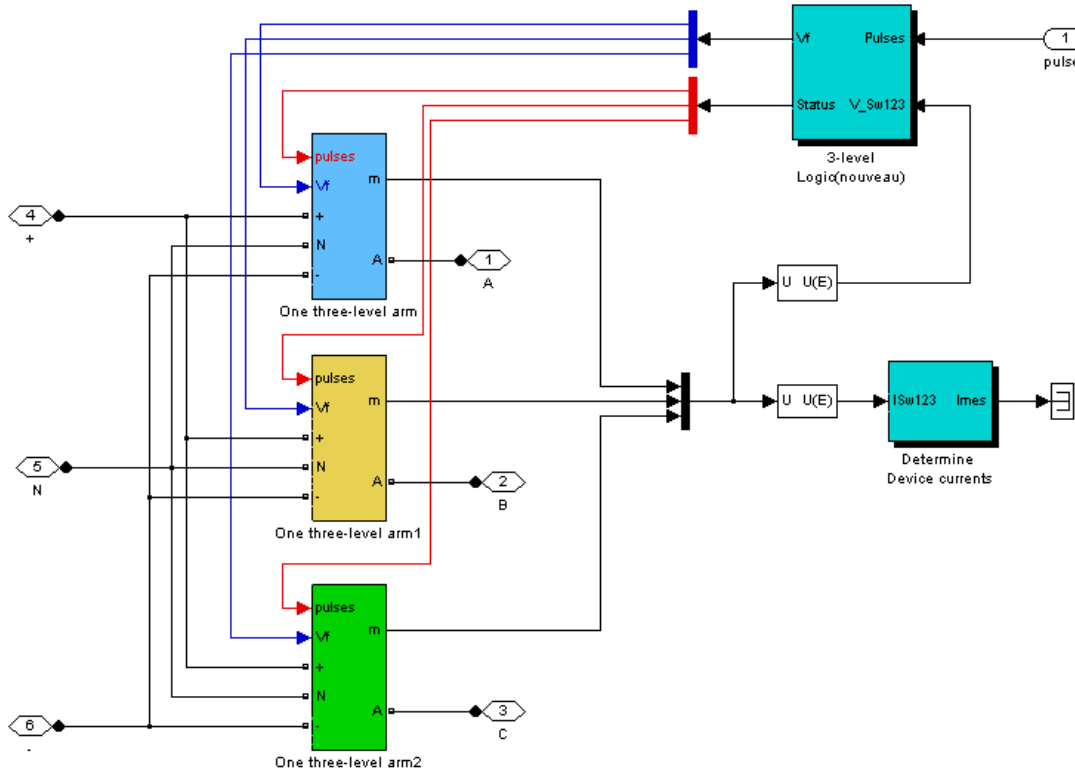
You can connect this line and leave the subsystem in this form. But, to clarify the connections and improve the subsystem's appearance, you can eliminate

the JunctionPoint blocks by using branching and reorganize the remaining blocks. You need to do the same editing steps for the other two One Arm Bridge subsystems.



You can now concentrate on the top-level diagram and connect its missing lines. You can again eliminate the JunctionPoint blocks and reorder the other

blocks as shown in the following diagram. Your 3-Level Bridge block is then fully updated to SimPowerSystems 3.0.



Index